

[illegible]


```
0001 0 MODULE SELVOL (  
0002 0     LANGUAGE (BLISS32),  
0003 0     IDENT = 'V04-001'  
0004 0 ) =  
0005 1 BEGIN  
0006 1  
0007 1  
0008 1 *****  
0009 1 *  
0010 1 *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY  
0011 1 *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.  
0012 1 *  ALL RIGHTS RESERVED.  
0013 1 *  
0014 1 *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
0015 1 *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
0016 1 *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
0017 1 *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
0018 1 *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
0019 1 *  TRANSFERRED.  
0020 1 *  
0021 1 *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
0022 1 *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
0023 1 *  CORPORATION.  
0024 1 *  
0025 1 *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
0026 1 *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.  
0027 1 *  
0028 1 *  
0029 1 *****  
0030 1  
0031 1 ++  
0032 1  
0033 1 FACILITY: F11ACP Structure Level 2  
0034 1  
0035 1 ABSTRACT:  
0036 1  
0037 1     This routine selects a suitable volume for the creation of a file  
0038 1     or the continuation of a file on some other volume.  
0039 1  
0040 1 ENVIRONMENT:  
0041 1  
0042 1     STARLET operating system, including privileged system services  
0043 1     and internal exec routines.  
0044 1  
0045 1 --  
0046 1  
0047 1  
0048 1 AUTHOR: Andrew C. Goldstein, CREATION DATE: 21-Nov-1978 16:59  
0049 1  
0050 1 MODIFIED BY:  
0051 1  
0052 1     V04-001 ACG0464 Andrew C. Goldstein, 7-Sep-1984 17:22  
0053 1     Rework to function in a cluster and process based environment  
0054 1  
0055 1     V03-002 ACG0407 Andrew C. Goldstein, 19-Mar-1984 14:53  
0056 1     Dispose of GETACC routine  
0057 1
```

SELVOL
V04-001

K 1
16-Sep-1984 01:09:23 VAX-11 Bliss-32 V4.0-742 Page 2
14-Sep-1984 12:30:46 DISK\$VMSMASTER:[F11X.SRC]SELVOL.B32;3 (1)

```
.. 58      0058 1 | V03-001 CDS0001      Christian D. Saether      2-Jan-1984
.. 59      0059 1 |      Use L_NORM linkage and BIND_COMMON macro.
.. 60      0060 1 |
.. 61      0061 1 | B0104  ACG0082      Andrew C. Goldstein,      8-Nov-1979  22:25
.. 62      0062 1 |      Skip over write locked volumes
.. 63      0063 1 |
.. 64      0064 1 | B0103  ACG0071      Andrew C. Goldstein,      12-Oct-1979  10:58
.. 65      0065 1 |      Range check placement RVN in volume selection
.. 66      0066 1 |
.. 67      0067 1 | B0102  ACG0039      Andrew C. Goldstein,      16-May-1979  13:02
.. 68      0068 1 |      Do correct error exit on contig allocation failure
.. 69      0069 1 |
.. 70      0070 1 | B0101  ACG0008      Andrew C. Goldstein,      26-Dec-1978  18:32
.. 71      0071 1 |      Add placement control support
.. 72      0072 1 |
.. 73      0073 1 | **
.. 74      0074 1 |
.. 75      0075 1 |
.. 76      0076 1 | LIBRARY 'SYSS$LIBRARY:LIB.L32';
.. 77      0077 1 | REQUIRE 'SRC$:FCPDEF.B32';
```

```
79 1068 1 GLOBAL ROUTINE SELECT_VOLUME (FIB, BLOCKS_NEEDED) : L_NORM NOVALUE =
80 1069 1
81 1070 1 ++
82 1071 1
83 1072 1 FUNCTIONAL DESCRIPTION:
84 1073 1
85 1074 1 This routine scans the RVT for the volume with the most free space,
86 1075 1 or, if a contiguous allocation is asked for, the volume with the
87 1076 1 most free space and sufficient contiguous space.
88 1077 1
89 1078 1
90 1079 1 CALLING SEQUENCE:
91 1080 1 SELECT_VOLUME (ARG1, ARG2)
92 1081 1
93 1082 1 INPUT PARAMETERS:
94 1083 1 ARG1: address of user FIB
95 1084 1 ARG2: number of blocks to be allocated
96 1085 1
97 1086 1 IMPLICIT INPUTS:
98 1087 1 LOC_RVN: placement RVN or 0
99 1088 1 CURRENT_VCB: VCB of current volume
100 1089 1
101 1090 1 OUTPUT PARAMETERS:
102 1091 1 NONE
103 1092 1
104 1093 1 IMPLICIT OUTPUTS:
105 1094 1 CURRENT_UCB, CURRENT_VCB, CURRENT_RVN: set to volume switched to
106 1095 1 UNREC_COUNT, UNREC_BLOCKS: count and LBN of blocks preallocated, if any
107 1096 1
108 1097 1 ROUTINE VALUE:
109 1098 1 NONE
110 1099 1
111 1100 1 SIDE EFFECTS:
112 1101 1 context switched to new volume, blocks may be allocated
113 1102 1
114 1103 1 --
115 1104 1
116 1105 2 BEGIN
117 1106 2
118 1107 2 MAP
119 1108 2 FIB : REF BBLOCK; ! user FIB arg
120 1109 2
121 1110 2 LOCAL
122 1111 2 STATUS, ! error status to return
123 1112 2 BEST_SIZE, ! largest volume of current scan
124 1113 2 BEST_RVN, ! RVN of above volume
125 1114 2 TRIED_IT : BITVECTOR [256], ! vector of volumes tried so far
126 1115 2 RVT : REF BBLOCK, ! address of relative volume table
127 1116 2 UCB : REF BBLOCK, ! UCB under consideration
128 1117 2 VCB : REF BBLOCK; ! VCB under consideration
129 1118 2
130 1119 2 BIND_COMMON;
131 1120 2
132 1121 2 EXTERNAL ROUTINE
133 1122 2 ALLOCATION_LOCK : L_NORM, ! acquire volume lock
134 1123 2 SWITCH_VOLUME : L_NORM, ! switch context to new volume
135 1124 2 ALLOC_BLOCKS : L_NORM; ! allocate blocks from storage map
```

```
136 1125 2
137 1126 2
138 1127 2
139 1128 2
140 1129 2
141 1130 2
142 1131 2
143 1132 2
144 1133 2
145 1134 2
146 1135 2
147 1136 2
148 1137 2
149 1138 2
150 1139 2
151 1140 2
152 1141 2
153 1142 2
154 1143 2
155 1144 2
156 1145 2
157 1146 2
158 1147 2
159 1148 2
160 1149 2
161 1150 2
162 1151 2
163 1152 2
164 1153 2
165 1154 2
166 1155 4
167 1156 4
168 1157 3
169 1158 4
170 1159 4
171 1160 4
172 1161 4
173 1162 5
174 1163 5
175 1164 5
176 1165 5
177 1166 5
178 1167 5
179 1168 6
180 1169 6
181 1170 6
182 1171 5
183 1172 4
184 1173 3
185 1174 3
186 1175 2
187 1176 2
188 1177 2
189 1178 2
190 1179 3
191 1180 3
192 1181 4

! We scan the volumes of the volume set in reverse size order. If a non-
! contiguous allocation is being done, we simply return with the volume with
! the most free space. If a contiguous request is made, try to do the allocation
! on each volume until it succeeds. The first pass (J = 0) is used to
! process RVN placement, if given.

ALLOCATION LOCK ();
RVT = .CURRENT_VCB[VCBSL_RVT];
IF .RVT EQL .CURRENT_UCB THEN RETURN; ! noop if not a volume set
IF .LOC_RVN GTRU .RVT[RVT$B_NVOLS] ! discard garbage RVN's
THEN LOC_RVN = 0;
CH$FILL (0, 256/8, TRIED_IT);
INCR J FROM (.LOC_RVN EQL 0) TO .RVT[RVT$B_NVOLS]
DO
  BEGIN
    BEST_SIZE = 0;
    BEST_RVN = 0;
    ! The inner loop scans the RVT for the volume (mounted) with the most free
    ! which we haven't tried yet. We take out the allocation lock on each
    ! volume before looking at it (by calling SWITCH_VOLUME) to get an up to
    ! date copy of the volume's free space.
    INCR K FROM (IF .J EQL 0 THEN .LOC_RVN ELSE 1)
      TO (IF .J EQL 0 THEN .LOC_RVN ELSE .RVT[RVT$B_NVOLS])
    DO
      BEGIN
        UCB = .VECTOR [RVT[RVT$B_UCBLST], .K-1];
        IF .UCB NEQ 0
        THEN
          BEGIN
            VCB = .UCB[UCBSL_VCB];
            SWITCH_VOLUME (.R);
            IF .VCB[VCBSL_FREE] GTRU .BEST_SIZE
            AND NOT .TRIED_IT[K]
            THEN
              BEGIN
                BEST_SIZE = .VCB[VCBSL_FREE];
                BEST_RVN = .K;
              END;
            END;
          END;
        END;
      END;
    ! Having picked a volume, check it for usefulness. A size of zero means the
    ! whole volume set is full. If we are trying for contiguous space, check if
    ! there is at least that much space and try the allocation.
    TRIED_IT[BEST_RVN] = 1;
    IF (
```

```
193 1182 4 IF .FIB[FIB$V_ALCON]
194 1183 4 THEN .BEST_SIZE LSSU .BLOCKS_NEEDED
195 1184 4 ELSE .BEST_SIZE EQL 0
196 1185 4 )
197 1186 3 THEN
198 1187 4 BEGIN
199 1188 4 IF .J NEQ 0
200 1189 4 THEN EXITLOOP;
201 1190 4 END
202 1191 4
203 1192 3 ELSE
204 1193 4 BEGIN
205 1194 4 SWITCH VOLUME (.BEST_RVN);
206 1195 4 UNREC_RVN = .BEST_RVN;
207 1196 3 IF (
208 1197 3 IF .BLOCKS_NEEDED NEQ 0
209 1198 3 THEN ALLOC_BLOCKS (.FIB, .BLOCKS_NEEDED, UNREC_LBN, UNREC_COUNT)
210 1199 3 ELSE 1
211 1200 3 )
212 1201 4 THEN RETURN;
213 1202 4 END;
214 1203 4
215 1204 4 LOC_RVN = 0; ! discard placement after first try
216 1205 4 LOC_LBN = 0;
217 1206 4 END; ! end of outer retry loop
218 1207 4
219 1208 4 ! We exit or fall out of the loop if we have tried all volumes in the set
220 1209 4 ! that seemed worth trying, and couldn't get anything.
221 1210 4
222 1211 4
223 1212 4 ERR_EXIT (SS$_DEVICEFULL);
224 1213 4
225 1214 1 END; ! end of routine SELECT_VOLUME
```

				OBFC 00000		
			SE	24	C2	00002
			5B	1C	AA	9E 00005
		0000G	CF		00	FB 00009
			50	98	AA	D0 0000E
			56	20	A0	D0 00012
		94	AA		56	D1 00016
					01	12 0001A
						04 0001C
6B	0B	A6	08		00	ED 0001D 1\$:
					02	1E 00023
					6B	D4 00025
20		00	6E		00	2C 00027 2\$:

		.TITLE	SELVOL
		.IDENT	\V04-001\
		.EXTRN	ALLOCATION_LOCK
		.EXTRN	SWITCH_VOLUME, ALLOC_BLOCKS
		.PSECT	\$CODE\$,NOWRT,2
		.ENTRY	SELECT_VOLUME, Save R2,R3,R4,R5,R6,R7,R8,- R9,R11 : 1068
		SUBL2	#36, SP
		MOVAB	28(BASE), R11 : 1117
		CALLS	#0, ALLOCATION_LOCK : 1134
		MOVL	-104(BASE), R0 : 1135
		MOVL	32(0), RVT
		CMPL	RVT, -108(BASE) : 1136
		BNEQ	1\$
		RET	
		CMPZV	#0, #8, 11(RVT), (R11) : 1138
		BGEQU	2\$
		CLRL	(R11) : 1139
		MOVC5	#0, (SP), #0, #32, TRIED_1T : 1141

		04	AE	0002C		CLRL	R4	1143
		54	D4	0002E		TSTL	(R11)	
		6B	D5	00030		BNEQ	3\$	
		02	12	00032		INCL	R4	
	6E	0B	A6	9A 00036 3\$:		MOVZBL	11(RVT), (SP)	
		54	D7	0003A		DECL	J	
		0090	31	0003C		BRW	16\$	
		57	7C	0003F 4\$:		CLRQ	BEST_RVN	1147
		50	D4	00041		CLRL	R0	1155
		54	D5	00043		TSTL	J	
		07	12	00045		BNEQ	5\$	
	52	50	D6	00047		INCL	R0	
		6B	D0	00049		MOVL	(R11), R2	
		03	11	0004C		BRB	6\$	
	52	01	D0	0004E 5\$:		MOVL	#1, R2	
	05	50	E9	00051 6\$:		BLBC	R0, 7\$	1156
	59	6B	D0	00054		MOVL	(R11), R9	
		04	11	00057		BRB	8\$	
	59	0B	A6	9A 00059 7\$:		MOVZBL	11(RVT), R9	
		52	D7	0005D 8\$:		DECL	K	1155
		24	11	0005F		BRB	10\$	
	55	40	A642	D0 00061 9\$:		MOVL	64(RVT)[K], UCB	1159
		1D	13	00066		BEQL	10\$	1160
	53	34	A5	D0 00068		MOVL	52(UCB), VCB	1163
		52	DD	0006C		PUSHL	K	1164
	0000G	CF	01	FB 0006E		CALLS	#1, SWITCH VOLUME	
		58	40	A3 D1 00073		CMPL	64(VCB), BEST_SIZE	1165
			0C	1B 00077		BLEQU	10\$	
07	04	AE	52	E0 00079		BBS	K, TRIED_IT, 10\$	1166
		58	40	A3 D0 0007E		MOVL	64(VCB), -BEST_SIZE	1169
		57	52	D0 00082		MOVL	K, BEST_RVN	1170
D8		52	59	F3 00085 10\$:		AOBLEQ	R9, K, 9\$	1155
00	04	AE	57	E2 00089		BBSS	BEST_RVN, TRIED_IT, 11\$	1180
		50	04	AC D0 0008E 11\$:		MOVL	FIB, -R0	1182
		08	16	A0 E9 00092		BLBC	22(R0), 12\$	
		AC	58	D1 00096		CMPL	BEST_SIZE, BLOCKS_NEEDED	1183
			0C	1E 0009A		BGEQU	14\$	
			04	11 0009C		BRB	13\$	
			58	D5 0009E 12\$:		TSTL	BEST_SIZE	1184
			06	12 000A0		BNEQ	14\$	
			54	D5 000A2 13\$:		TSTL	J	1188
			24	13 000A4		BEQL	15\$	
			2D	11 000A6		BRB	17\$	1189
			57	DD 000A8 14\$:		PUSHL	BEST_RVN	1194
	0000G	CF	01	FB 000AA		CALLS	#1, SWITCH VOLUME	
	2C	AA	57	D0 000AF		MOVL	BEST_RVN, Z4(BASE)	1195
			08	AC D5 000B3		TSTL	BLOCKS_NEEDED	1197
			21	13 000B6		BEQL	18\$	
			28	AA 9F 000B8		PUSHAB	40(BASE)	1198
			24	AA 9F 000BB		PUSHAB	36(BASE)	
		7E	04	AC 7D 000BE		MOVQ	FIB, -(SP)	
	0000G	CF	04	FB 000C2		CALLS	#4, ALLOC_BLOCKS	
		0F	50	E8 000C7		BLBS	R0, 18\$	
			6B	D4 000CA 15\$:		CLRL	(R11)	1204
			20	AA D4 000CC		CLRL	32(BASE)	1205
			6E	F1 000CF 16\$:		ACBL	(SP), #1, J, 4\$	1143

FF6A

54

01

SELVOL
V04-001

C 2
16-Sep-1984 01:09:23
14-Sep-1984 12:30:46

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[F11X.SRC]SELVOL.B32;3
Page 7
(2)

0850 8F BF 000D5 17% CHMU #2128
04 000D9 18% RET

: 1212
: 1214

: Routine Size: 218 bytes. Routine Base: \$CODE\$ + 0000

: 226 1215 1
: 227 1216 1 END
: 228 1217 0 ELUDOM

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	218	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	24	0	1000	00:01.9

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:SELVOL/OBJ=OBJ\$:SELVOL MSRC\$:SELVOL/UPDATE=(ENH\$:SELVOL)

: Size: 218 code + 0 data bytes
: Run Time: 00:18.1
: Elapsed Time: 00:35.6
: Lines/CPU Min: 4032
: Lexemes/CPU-Min: 47463
: Memory Used: 223 pages
: Compilation Complete

0173 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

SCHFCB
LIS

SND5MB
LIS

SHFDIR
LIS

SNDER
LIS

TRUNC
LIS

FAL

FAL
MAP

SELVOL
LIS

DAPDEF
MOL

SMALOC
LIS

SNOBAD
LIS

SWTTL
LIS

WITURN
LIS